Prof. Ghassan Alregib
School of Electrical and Computer Engineering
Georgia Institute of Technology
912-966-7937 | gregib@ece.gatech.edu
http://users.ece.gatech.edu/gregib

HP FanCamera in Distant Learning
Project Report – October 26, 2007

Summary:
The HP FanCamera was received in May 2007 from HP. During summer 2007, the camera was placed and installed in an experimental classroom at Georgia Tech Savannah Campus. We have designed and built a network delivery framework with a graphical user interface. The camera now is operational and remote students could connect to it and interact with it by zooming and panning.
This report summarizes our accomplishments and the ongoing work on the HP fancamera in a classroom environment. The report is divided into five sections. The first section describes the setup in our experimental classroom. The second section describes the client-sever based network application that we have built for the fan camera. The third section describes the client enhanced interactivity application that utilizes the client's GPU to digitally manipulate the received image. The fourth section describes our graphical user interface. And finally the last section summarizes our current work.

Setup:
We have installed the camera in an experimental classroom (PARB Room #212) looking at a 32' X 8' wall. The camera is placed at a distance 18 ft away from the whiteboard. In order to have the camera looking at the top of the whiteboard we designed a setup using metallic racks that accommodate the camera, the concentrators, the power adapters, and the computer. The camera resides on a tripod on the top of the highest shelf as shown if Figure 1. The setup is arranged so that all kind of cable interferences are minimized and, henceforth, maximize the reliability of the camera. Also, we have setup all the needed software suits on the powerful 4 GB dual processor machine with a 256 MB GPU (will be upgraded to 768MB by November 2007).

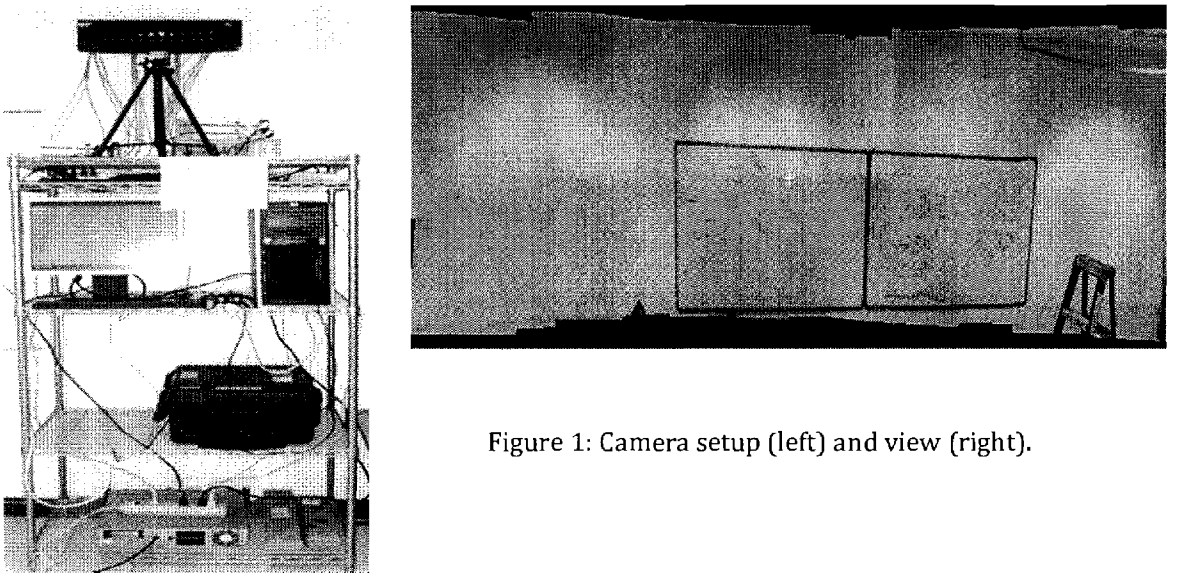

Figure 1: Camera setup (left) and view (right).

Client-Server Framework:
We have built a client-server application that allows students to connect over the internet and receive a real time video of the whiteboard on their computers. Currently, we can have up to 25 students connecting simultaneously to our server. The framework can be divided into a registration, request, and broadcast process.

1. *Registration (server):*
    - The server registration specifies the number of allowed connections to the server (currently 25 maximum).

- In order to be able to connect to the server, each client should be registered using an IP address and a client ID (in this case we used the IP address as the client ID).
- Once the students are registered and the number of connections is specified the server starts listening to connection requests.

2. *Request (Client):*
   - A student initializes a connection using the server IP address and an available bit rate.
   - The client application sends a connection request with client ID and its advertised bit rate.
   - Client starts listening to a video stream.
3. *Broadcast (Server):*
   - Server application listening for requests receives a connection request.
   - Server checks the client ID:
     - If the ID is available and the number connections did not exceed the specified number, it performs mpeg2 compression at the received rate and sends video.
     - Else, it ignores the request.
   - The video stream is currently compressed using MPEG-II and a current effort is undergone to employ H.264.

Client End Interactivity:

An interactive desktop application is demonstrated in which the student can digitally manipulate a virtual camera with pan, tilt, zoom, roll, and save. The interactivity is supported a both ends at the server end and the client end. At the server (camera) end the changes on the camera view are copied on all out going connections. At the client end each student can further manipulated the received image. Currently the client utilizes the GPU to operate on the received image, and we are working on a different approach in which compositing can be done a client side to enhance the zoom experience. The application that we have built is also friendly to users that do not have powerful GPUs with limited interactivity options.

Graphical User Interface:

We have built a graphical user interface shown in figure 2. Using the control panel of this GUI, a student can specify the server IP address, choose to enable the local interactivity, and choose to save/record the video. Once the student hits the connect button the received video will show up on a separate window. The student can operate on the received video if the interactivity is enabled. The video statistics such as PSNR, frame rate and other related info will show in the statistics box as well. The graphical user interface will be extended to support scene selection, in which images from 24 cameras will be shown and the student will select images of focus only

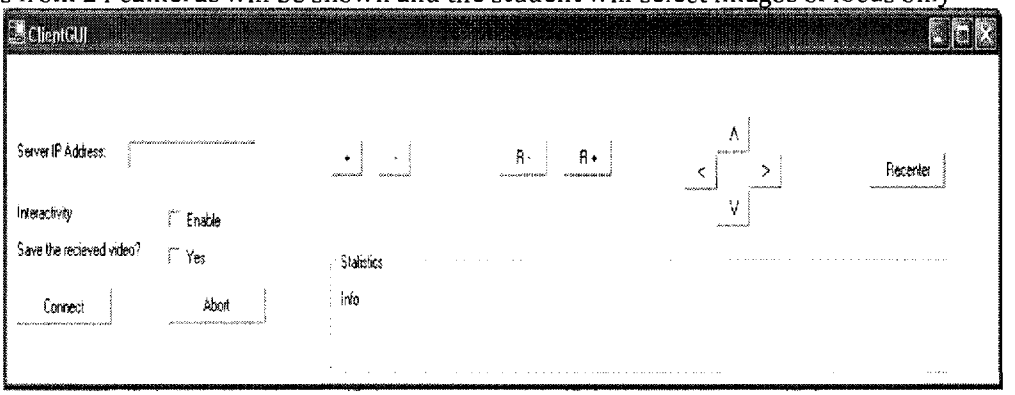

Figure 2: the client's graphical user interface. Only the control panel is shown.

Future work:

Currently we are experimenting on streaming individual imagers and then composting at the client end. The compression will also be upgraded to H.264 and then quality assessment will be conducted on the resulting streaming experience. Automatic zooming and tracking will also be considered along with client server streaming adjustment on the streamed images.